REMARKS

Claims 10, 41-47, 55, and 57-61 are pending in the application.

Claims 10, 41-47, 55, and 57-61 are currently amended, and claims 48-54 and 56 are canceled. Applicants respectfully submit that no new matter is added to currently amended claims 10, 41-47, 55, and 57-61.

Claim 10 stands rejected under 35 U.S.C. §102(e) as anticipated by U.S. Patent No. 6,257,774 to Stack.

Claims 41-61 stand rejected under 35 U.S.C. §102(e) as anticipated by U.S. Patent No. 6,792,618 to Bendinelli.

Applicants respectfully traverse the rejections based on the following discussion. The following paragraphs are numbered for ease of future reference.

I. The Prior Art Rejections

A. The 35 U.S.C. § 102(e) Rejection over Stack

1. The Stack Reference

[0001] Stack discloses, "The present invention is generally related to an automated program and documentation generating system and, in particular, to a multi-stage, rule based system for managing the concurrent development of a program and documentation specifications that are then processed into program code and directly corresponding user documentation." (col. 1, lines 9-15).

[0002] Stack also discloses, "The skill rules 20 represent the low-level functional operations that are utilized to effectively implement the application of the program rules. That is, the skill rules provide basic functional operations that are widely used in the generation of an intermediate representation of a generated program. Although any number of skill rules 20 may be defined for use within the system 10, the present implementation provides for eight basic skills including: 1) Single Line ... ; 2) Recurring Line ... ; 3) Variable List ... ; 4) Recurring List ... ; 5) Implied List ... ; 6) Translate ... ; 7) Combined List ... ; and 8) Mixed List ... . The program rules 18 may exist in the program rule base in a seemingly arbitrary order. However, the program rules 18 may be considered to be ordered when viewed as applicable to a particular program type and sub-type. Thus, for a particular program sub-type, program rules will appear to be ordered in an evaluation sequence that, upon application to the application structure and sequences 12, will provide for the realization of a corresponding application program sub-type. The processing of program rules 18 begins at a logical root rule and proceeds through the ordered set of rules that are either generic to all program types or specific to the sub-program type presently being evaluated. As the program rules 18 are being evaluated against the application structure and sequences 12, the skill rules 20 are executed to develop a logically complete representation of the application program to be realized through the generation of the code 24. That is, the skill rules provide for the instantiation of the logical statements that, when further evaluated against the syntax rules 22, provide for the generation of the code 24. Since each logical statement of an application program can be fully represented as a constant or variable element, represented either in numerical or ASCII form, or list structures of such elements, even the limited set of eight basic skills is sufficient to support a full and functionally complete set of program rules 18 necessary to generate most, if not all, accounting or accounting related applications. The eight basic skill rules are also quite sufficient to generically support generation of many other types of applications as well. Should either different functionality be desired or required for a particular application or a higher level program rule be desired to be supported through the construction of a complex skill, such skills may be readily added to the skill rule base 20." (col. 6, line 17 to col. 7, line 47).

[0003] Stack further discloses, "The image editor 32 also provides for image manipulation functions, including clipping, cutting, pasting and selecting sections of the image page. These manipulation operations permit image portions to be moved, replicated, and deleted. Based on user input, these operations are implemented with the appearance of a conventional block operation. However, the image editor 32 provides, as part of block operation, for the hidden selection of the attributes associated with each of the display items and fields within the blocked portion of the image. The various block object operations are preferably implemented subject to the evaluation of block operation rules retrieved from the knowledge base 34. These rules aid in the reasonable selection of the attributes associated with a blocked object and their merger with existing attributes when an object is copied back into an image. As such, the block operation also provide for consistency check and merge validation in the execution of the image editor 32." (col. 9, line 54 to col. 10, line 3).

[0004] Stack yet further discloses, "Block portions of the image may also be copied to an object store database 42 and stored subject to retrieval upon reference to a uniquely assigned object identifier. Since the image editor 32 preferably associates attribute identifiers with each display item or field rather than just references to an attribute, objects stored to the object store database 42 are logically self-contained and do not inherit subsequent changes made through the image editor 32 to any original or other copy of the object. Consequently, the objects stored by the object store database 42 are essentially static, though the objects may be dynamically selected with arbitrary size and complexity. As display items and field data are input into the image editor 32 and attributes are assigned, the image editor 32 derives and provides definitional information to an application sequence generator that produces a set of application sequences that comprehensively describe a program segment as well as identify the database program and the display item or field, the location of the item or field within the space of the image page, and each of the functional attributes associated with the item or field. These application sequences are then provided to the application sequences stacks 38." (col. 10, lines 4-26).

[0005] Stack yet further discloses, "Once the application structure 36 and application sequences 38 for a program have been constructed through the operation of the image editor 32, an application author 44 is invoked to generate a corresponding set of compliable or interpretable code. The application author operates from the application structure 36, application sequences 38, the contents of the help file 40 and a corresponding set of programs, skill and syntax rules as stored by the knowledge base 34. The application author 44, operating generally as an expert system, applies program, skill and syntax rules to the available definition of the application structure 36 in concert with the specific application sequences 38 to produce the code for a program of the particular type and sub-type being generated. That is, a program of the selected program type and sub-type is constructed by the operation of the expert system of the author 44 based on the defined application structure 36 and available applications sequences 38. In accordance with the present invention, the code produced by the author 44 is the product of the program rules that define a particular program type and sub-type to the extent that such rules are consistent with the provided application structure 36 and application sequences 38. Since the code statements are effectively independently generated by the operation of the author 44 based on the application structure and sequences 36, 38 the individual code statements are inherently unified with respect to the number, naming and typing of all variables instantiated by the author 44. No further unification of variables or mediation of the interfaces between segments of a single program is required. Interfaces between segments, to the extent that they are required, are automatically matched by the operation of the expert system of the author 44 based on the definition of the application structure 36 and the application sequences that define data items and fields that are shared by different segments of a common program." (col. 10, lines 27-61).

[0006] Stack yet further discloses, "The system variables thus provide an easy mechanism allowing for the testing of current state, the manipulation and conversion or translation of data between records and fields and between sets of fields, for the establishment of option sets, and in specifying logic function operations for particular fields or actions to be taken entering or exiting fields. While the system variables are available for reference in feature packets, they are more commonly used in or referenced by the program and test rules implemented in the knowledge base 34. Most of the system variables affect or detail lower level functions relative to the functional specifications established in relation to feature packets. For example, a mere field name reference in a feature packet that supports editing of the displayed value will be implicitly evaluated subject to the program and test rules to determine if an initial default value is to be displayed in the field. The test rules that will determine if a default value is to be displayed and the program rules that will determine or fetch the default value can reference the system variables extensively in implementing the necessary functional logic. If a field is to be defaulted to the current date, but permitted to be user modifiable, then the rules may retrieve the current date (!Data Date) and place the date in the field (!Display To) if the current field (!a) is empty. Consequently, a direct and highly non-procedural capability is provided by the use of the system variables within function packets, while providing a detailed yet highly abstracted capability in support of the formation of program and test rules." (col. 25, lines 18-45).

[0007] Stack yet further discloses, "Frames 112 of the application sequences are generally sequentially evaluated by the expert system underlying the application author 44. As indicated in FIG. 7, programmer specifications, representing the collected application sequences are received as an input to the application author in combination with the application structure 36 as represented by the file definitions 92 and program definitions 94. In addition, the application author 44 has access to the knowledge base 34 for retrieval of program rules 120, test rules 122, basis rules 124, skill rules 126, and syntax rules 128. Based on the program type and sub-type obtained effectively from the program definition 94, the applicable set of program rules 120 and corresponding test rules 122 are selected for evaluation by the author 44 against the programmer specifications. These selected program rules anticipate and impose a general framework to the functions and relationships presented by the programmer specifications. As a consequence, the program rules 120 serve to select out of the programmer specifications the information from the available application sequences needed to satisfy the program rules 120. The test rules 122 serve as high level qualifications tests to determine whether different subsets of the program rules 120 are to be utilized in view of the particular programmer specifications being supplied to the application author 44. Consequently, the comprehensive structure and flow of a particular code module generated by the application author 44 is inferentially dependant on the fields and field relationships initially established based on user input." (col. 29, lines 34-62).

2. Argument

[0008] Stack merely discloses an automated program and documentation generating system that uses a multi-stage, rule based system for managing the concurrent development of a program and documentation specifications, which are then processed into program code and directly corresponding user documentation. (col. 1, lines 9-15).

[0009] Currently amended, independent claim 10 clearly describes at least the following features of the invention:

"wherein:

a rule comprises a test and an action ... ,

said condition corresponding to: if, at time t_i, a specific image of a presentation sequence is presented,

said action comprising a sequence of operations applied to said presentation sequence, wherein said presentation sequence comprises a sequence of static objects, each of said static objects ... comprising any of an image, text, and a video frame ... ;

...

outputting, by said computer, said previously-generated presentation sequence that is modified to a monitor for display." (emphases added).

[0010] That is, the invention provides a method for outputting a modified previously-generated presentation sequence, which comprises any of a modified sequence of images, text, and video frames, for display by a monitor.

[0011] In contrast, Stack merely discloses an automated program and documentation generating system that uses a multi-stage, rule based system for managing the concurrent development of a program and documentation specifications, which are then processed into program code and directly corresponding user documentation. (col. 1, lines 9-15).

[0012] Applicants respectfully submit that nowhere does Stack disclose, teach or suggest outputting a modified previously-generated presentation sequence, which comprises any of a modified sequence of images, text, and video frames, for display by a monitor, as clearly disclosed and claimed by the invention.

[0013] For at least the reasons outlined above, Applicants respectfully submit that nowhere does Stack disclose, teach or suggest at least the invention's features of: " wherein: a rule comprises a test and an action ... , said condition corresponding to: if, at time t_i, a specific image of a presentation sequence is presented, said action comprising a sequence of operations applied to said presentation sequence, wherein said presentation sequence comprises a sequence of static objects, each of said static objects ... comprising any of an image, text, and a video frame ... ; ... outputting, by said computer, said previously-generated presentation sequence that is modified to a monitor for display", as recited in currently amended, independent claim 10. Accordingly, Stack does not anticipate the subject matter of currently amended, independent claim 10 under 35 U.S.C. §102(e). Withdrawal of the rejection of claim 10 under 35 U.S.C. §102(e) as anticipated by Stack is respectfully solicited.

B. The 35 U.S.C. § 102(e) Rejection over Bendinelli

1. The Bendinelli Reference

[0014] Bendinelli discloses, "Uniform Resource Locators (URLs) or other network information identifiers embedded in television signals are processed in order to permit viewers to customize the display of a corresponding program. In an illustrative embodiment, at a time prior to the scheduled display of a program or portion thereof, a given viewer makes a selection of one of a number of available alternate characteristics for the program, such as one of a number of available alternate endings. At a later time, a base URL embedded in the television signal is extracted and processed using the viewer selection in order to generate a combined URL. The combined URL is then used to establish a connection over a network with a corresponding web site or other information source, and information is retrieved from the web site and stored in a memory. The retrieved information is subsequently taken from the memory and used to modify the manner in which the program is displayed to the viewer. For example, the retrieved information may include video which is displayed to the viewer in place of a portion of the program." (Abstract).

[0015] FIG. 1 of Bendinelli discloses,

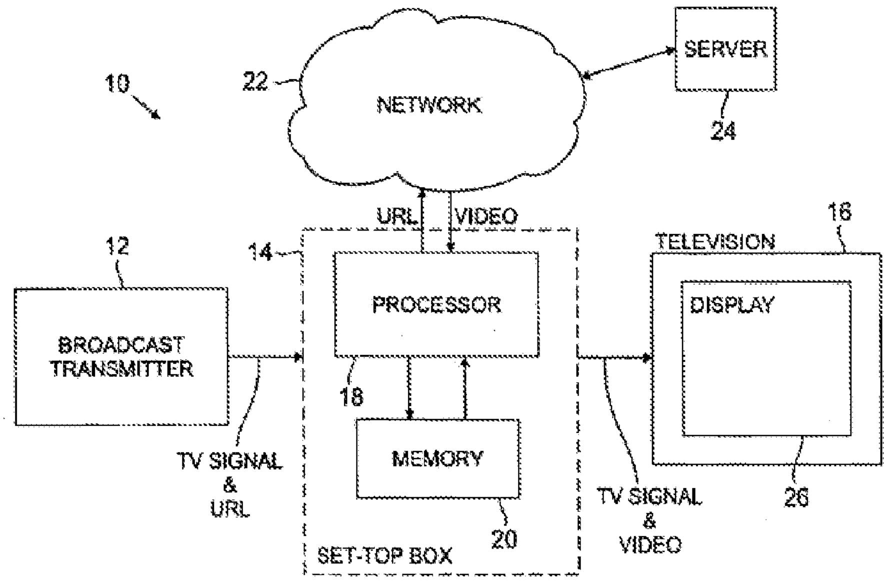
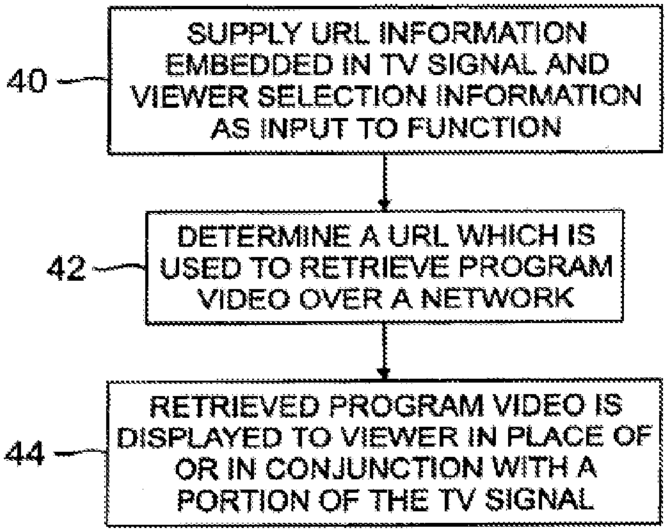


FIG. 1

[0016] Bendinelli also discloses, "FIG. 1 shows a system 10 for implementing viewer-customized display of transmitted programming in accordance with the invention. The system 10 includes a broadcast transmitter 12, a set-top box 14 and a television 16. A television signal transmitted from the broadcast transmitter 12 includes a program portion and one or more URLs. The URLs are transmitted so as to coincide with specified sections of the television programming represented by the program portion of the signal. For example, when the programming nears the ending of a particular program, the corresponding transmitted URL may identify a web site which stores video for several alternate endings to the program. Alternatively, near the beginning of a mystery show, information may be retrieved to obscure, or provide, various clues to the mystery's resolution." (col. 3, lines 30-45).

[0017] Bendinelli further discloses, "As will be described in greater detail in conjunction with FIG. 2 below, a function implemented in the set-top box 14 utilizes viewer selection information stored in the memory 20 to modify the extracted or "base" URL, in order to identify a particular viewer-selected program characteristic available from the web site of the base URL. In this manner, the set-top box 14 requests the retrieval of video corresponding to a viewer-selected program characteristic from the server 24. The requested video is then downloaded from the server 24 to the memory 20 of set-top box 14 via the established connection over the network 22." (col. 4, lines 21-32).

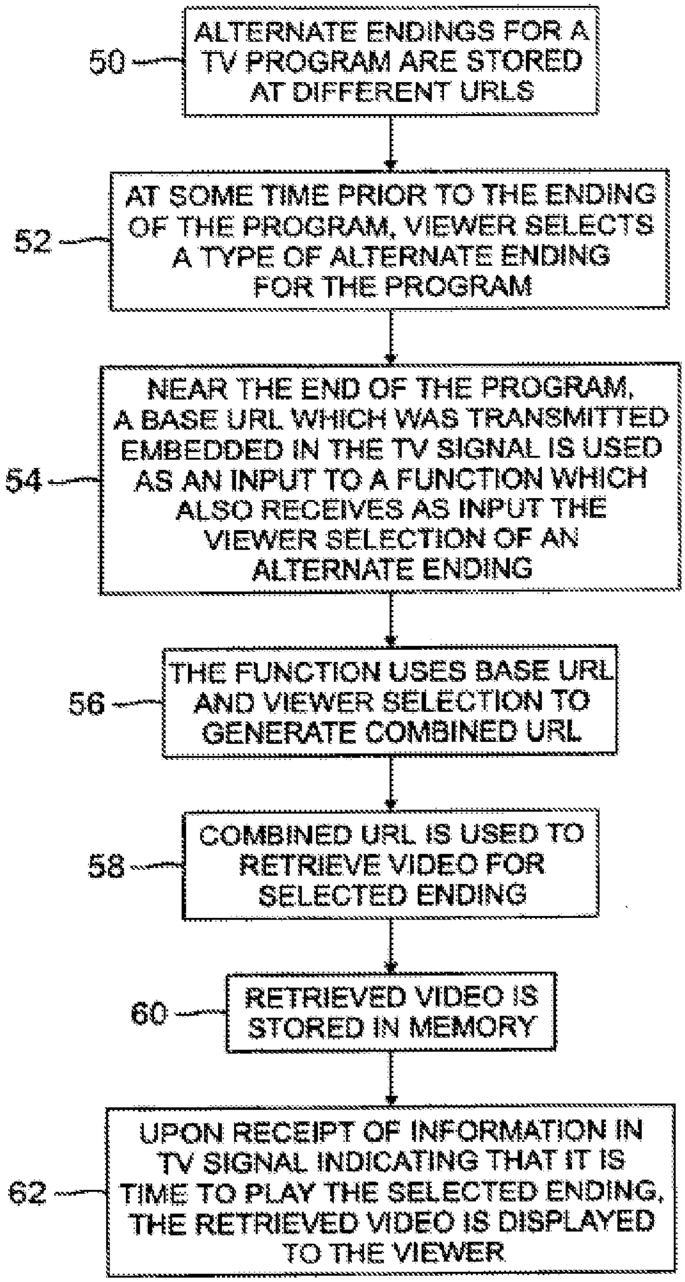
[0018] FIG. 2 of Bendinelli discloses,


40
SUPPLY URL INFORMATION EMBEDDED IN TV SIGNAL AND VIEWER SELECTION INFORMATION AS INPUT TO FUNCTION
42
DETERMINE A URL WHICH IS USED TO RETRIEVE PROGRAM VIDEO OVER A NETWORK
44
RETRIEVED PROGRAM VIDEO IS DISPLAYED TO VIEWER IN PLACE OF OR IN CONJUNCTION WITH A PORTION OF THE TV SIGNAL

[0019] Bendinelli yet further discloses, "FIG. 2 is a flow diagram illustrating the general operation of the system 10 of FIG. 1 in more detail. Step 40 indicates that URL information embedded in a television signal is extracted and used as an input to a retrieval control function. The URL information may be the above-noted base URL identifying a web site storing a number of alternative program characteristics. The function may be a URL processing function implemented by the processor 18 of set-top box 14. Step 40 further indicates that another input to the function is viewer selection information which is stored in the memory 20 of the set-top box 14. This viewer selection information identifies one or more program characteristics previously selected by a viewer. In step 42, the function uses the extracted URL information and the viewer selection information to determine a URL which is used to retrieve program video from server 24 over the network 22. The retrieved program video provides the program characteristic selected by the viewer." (col. 4, lines 47-65).

[0020] Bendinelli yet further discloses, "FIG. 3 is a flow diagram illustrating a more specific example of the operation of the system of FIG. 1. In this example, the program characteristic selected by a viewer is an ending for a program or a particular portion of a program, such as a happy ending or a sad ending. Step 50 indicates that a number of alternate endings for the program are stored at different UTRLs. At some time prior to the ending of the program, the viewer selects a type of available alternate ending for the program, as shown in step 52. This selection may be in response to available selection information displayed to the viewer at some time prior to the ending, or may be performed automatically based on a user profile. For example, a listing of available selections may be generated and displayed to the viewer by retrieving information identified by an additional URL which is transmitted embedded in the television signal. The viewer selection is stored in the memory 20 of the set-top box 14. In step 54, near the end of the program, a base URL, e.g., http://www.baseURL.com, which was transmitted embedded in the corresponding television signal, is extracted as previously described and applied as an input to a function implemented by processor 18 of set-top box 14. The function in step 54 also receives as an input the corresponding viewer selection of the type of available alternate ending he desires. In step 56, the function processes the base URL and the viewer selection to generate a combined URL. For example, the combined URL may be http://www.baseURL.com/happy, which corresponds to a stored video which when integrated with the broadcast television signal results in a happy ending for the program, or http://www.baseURL.com/sad, which corresponds to a stored video which when integrated with the broadcast television signal results in a sad ending for the program. In step 58, the combined URL is used to retrieve the corresponding video via a connection established over the network 22. The retrieved video is then stored in a memory quickly accessible by the set-top box 14, e.g., memory 20, as shown in step 60. Upon receipt of information which also is transmitted embedded in the television signal and indicates the presentation time for the ending, the set-top box 14 retrieves the corresponding video from memory 20 and causes it to be displayed to the viewer, such that the viewer is provided with the desired ending. As previously noted, the television signal may include a default ending which is overwritten in whole or in part by the retrieved video. This default ending may be used in the event the viewer does not select an alternate ending, or if there is a problem in retrieving the video from the combined URL generated in step 56 of FIG. 3." (col. 5, lines 8-55).

[0021] FIG. 3 of Bendinelli discloses,


50
ALTERNATE ENDINGS FOR A TV PROGRAM ARE STORED AT DIFFERENT URLS
52
AT SOME TIME PRIOR TO THE ENDING OF THE PROGRAM, VIEWER SELECTS A TYPE OF ALTERNATE ENDING FOR THE PROGRAM
54
NEAR THE END OF THE PROGRAM, A BASE URL WHICH WAS TRANSMITTED EMBEDDED IN THE TV SIGNAL IS USED AS AN INPUT TO A FUNCTION WHICH ALSO RECEIVES AS INPUT THE VIEWER SELECTION OF AN ALTERNATE ENDING
56
THE FUNCTION USES BASE URL AND VIEWER SELECTION TO GENERATE COMBINED URL
58
COMBINED URL IS USED TO RETRIEVE VIDEO FOR SELECTED ENDING
60
RETRIEVED VIDEO IS STORED IN MEMORY
62
UPON RECEIPT OF INFORMATION IN TV SIGNAL INDICATING THAT IT IS TIME TO PLAY THE SELECTED ENDING, THE RETRIEVED VIDEO IS DISPLAYED TO THE VIEWER

[0022] Bendinelli yet further discloses, "The computer 210 may implement a web browser program or other suitable network access software. The computer 210 uses the web browser program and an attached modem to establish a connection via network 116 to the server 120 corresponding to a given extracted URL. The server 120 delivers program video or other program information specified by the extracted URL to the computer 210 over the network 116, and the computer 210 stores the retrieved program information in the memory 214. At a time indicated by further information embedded in the television signal, e.g., subsequent to the embedded URL, the processor 212 directs delivery of the retrieved program information to a television 218 for presentation to a viewer on display 220. The function for determining a combined URL from an embedded base URL may be implemented in this embodiment in either the processor 212 or the tuner card 216 of the computer 210. As in the previous embodiments, the retrieved program information may display in a window 222 of the display 220, using size and location information embedded in the television signal, or overwritten on a portion of the original program portion of the television signal." (col. 7, lines 6-26).

2. Argument

[0023] Bendinelli merely discloses that Uniform Resource Locators (URLs) embedded in television signals may permit viewers to customize the display of a corresponding television program. (Abstract).

[0024] Currently amended, independent claims 41 and 55 clearly describe at least the following features of the invention:

"wherein:

each rule of said set of rules comprises a test and an action, said test specifying a condition for implementing said action,

said condition corresponding to: if, at time t_i, a specific image of a presentation sequence is presented,

said action comprising a sequence of operations applied to said presentation sequence, wherein said presentation sequence comprises a sequence of static objects, each of said static objects being assigned a relative time, t_k, in said presentation sequence and comprising any of an image, text, and a video frame". (emphases added).

[0025] That is, the invention provides a method for outputting a modified previously-generated presentation sequence of static objects, based upon a rule acting on a condition, defined as: if, at time t_i, a specific image of a presentation sequence is presented.

[0026] In contrast, Bendinelli address a method of customizing a video program with, for example, alternative endings. (Abstract). Nowhere does Bendinelli disclose a rule-based action requiring a specific image being present at a time t_i in a presentation sequence of static objects.

[0027] For at least the reasons outlined above, Applicants respectfully submit that Bendinelli does not disclose, teach or suggest at least the following features of the invention: "wherein: each rule of said set of rules comprises a test and an action, said test specifying a condition for implementing said action, said condition corresponding to: if, at time t_i, a specific image of a presentation sequence is presented, said action comprising a sequence of operations applied to said presentation sequence, wherein said presentation sequence comprises a sequence of static objects, each of said static objects being assigned a relative time, t_k, in said presentation sequence and comprising any of an image, text, and a video frame", as recited in currently amended, independent claims 41 and 55. Accordingly, Bendinelli does not anticipate the subject matter of currently amended, independent claims 41 and 55, and dependent claims 42-47 and 57-61 under 35 U.S.C. §102(e). The rejection of canceled claims 48-54 and 56 is moot. Withdrawal of the rejection of claims 41-61 under 35 U.S.C. §102(e) as anticipated by Bendinelli is respectfully solicited.

II. Formal Matters and Conclusion

Claims 10, 41-47, 55, and 57-61 are pending in the application.

With respect to the rejections of the claims over the cited prior art, Applicants respectfully argue that the present claims are distinguishable over the prior art of record. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections to the claims.

In view of the foregoing, Applicants submit that claims 10, 41-47, 55, and 57-61, all the claims presently pending in the application, are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest time possible.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary.

Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 50-0510.

Respectfully submitted,

Dated: June 3, 2011

/Peter A. Balnave/

Peter A. Balnave, Ph.D.
Registration No. 46,199

Gibb I.P. Law Firm, LLC
844 West St., Suite 100
Annapolis, MD 21401
Voice: (410) 705-6403
Fax: (410) 630-1656
Email: Balnave@gibbiplaw.com
Customer Number: 29154